

# Security-SLA-Guaranteed Service Function Chain Deployment in Cloud-Fog Computing Networks

Dongcheng Zhao, Long Luo, Hongfang Yu, Victor Chang,  
Rajkumar Buyya, and Gang Sun

**Abstract**—Network Function Virtualization (NFV) has gained prominence in next-generation cloud computing, such as the fog-based radio access network, due to their ability to support better QoS in network service provision. However, most of the current service function chain (SFC) deployment researches do not consider the Security-Service-Level-Agreement (SSLA) in the deployment solution. Therefore, in this work, we introduce the SSLA into SFC deployment to defend attacks. Firstly, we formulate the SSLA guaranteed SFC deployment problem by using linear programming. Then, we propose the Maximal-security SFC deployment algorithm (MS) to maximize the security of the SFC deployment. However, the MS algorithm results in a high deployment cost. To reduce the deployment cost, we propose the Minimal-cost and SSLA-guaranteed SFC deployment algorithm (MCSG) to minimize the deployment while satisfying the SSLA. In order to reduce the blocking ratio caused by MCSG, the Minimal-cost and SSLA-guaranteed SFC deployment algorithm with feedback adjustment (MCSG-FA) is proposed. Finally, we evaluate our proposed algorithms through simulations. The simulation results show that the blocking ratio and the deployment cost of our algorithms are better than that of the existing algorithm when meeting the SSLAs.

**Index Terms**— Service function chain; Network function virtualization; Deployment; Security; Fog-cloud computing

## I. INTRODUCTION

Virtualization is a key technology to improve the network flexibility [1-3]. As the development of network, the network function virtualization (NFV) technology has been proposed to transfer the traditional network function to the virtual network function (VNF) to improve the network resource utilization [4]. Multiple VNFs consist of service function chains (SFCs) to guarantee user's service strategy, and the SFC requests are deployed into the cloud network to provide services [5, 6].

With more and more users to use cloud network, the network delay and congestion are becoming more and more serious in the centralized cloud computing. Additionally, the

security of service is also being challenged. In order to solve these challenges, distributed fog computing is proposed to extend and supplement the cloud computing [7]. Recently, the fog-based radio access network is becoming a new research hotspot [8]. In fog radio access network (FRAN), there are some fog nodes which similar to cloud nodes. These fog nodes which provide services for users can be virtualized, thereby improving the flexibility of the networks [9]. However, the resource capacity of the fog node is usually less than that of the cloud node. Thus, joint use of cloud-fog computing can decrease network congestion and delay and thus provide high quality services for mobile users.

Currently, there are many researches focus on SFC deployment in cloud computing [10-13]. For example, in [13], Ricard Vilalta et al. conducted a software defined network (SDN)/NFV deployment experiment for 5G services based on cloud-fog computing to optimize the VNF deployment while satisfying the constraints (e.g., latency). With the increase of SFC requests, how to ensure the security-service-level-agreement (SSLA) of services has become a big challenge. In order to guarantee the SSLAs of services when the services are attacked, there are some studies on the security of NFV. For example, Mahdi Daghmehchi Firoozjaei et al. classified network security threats and proposed the possible solutions from the architectural layer [14]. These studies proposed some security architectures of NFV to ensure the SSLAs of services from the architecture layer but still cannot defend all attacks. Furthermore, these studies did not consider utilizing the federated environment of the cloud-fog network to provide more secure services for mobile users. In [15], the authors proposed the security-on-demand services for ATM networks, i.e., different user requests have different SSLA requirements. Then, the user request is deployed according to the user's SSLA requirement. Besides, in [16, 17], the authors discussed how to manage and meet users' SSLA demands. In commercial applications, some companies (e.g., Huawei) provide servers with different security levels to users to deploy service requests with different SSLA requirements, and the higher security level of the server, the higher charge.

To solve the challenge in management of SSLA in SFCs, in this paper, we consider that each service node and each physical link has a security possibility that can defend attacks. Therefore, we study the SFC placement problem in the federated environment of the cloud-fog network to meet the SSLA requirements of users when each service node and each physical link has a fixed security level. The main contributions of this paper are as follows.

- To resolve the SFC deployment problem with the SSLA

D. Zhao and L. Luo are with Key Lab of Optical Fiber Sensing and Communications (Ministry of Education), University of Electronic Science and Technology of China, Chengdu, China (e-mail: zhaodc11@gmail.com; longluo.uestc@gmail.com).

H. Yu is with Key Lab of Optical Fiber Sensing and Communications (Ministry of Education), University of Electronic Science and Technology of China, Chengdu, China; and is also with Peng Cheng Laboratory, Shenzhen, China (e-mail: yuhf@uestc.edu.cn)

V. Chang is with School of Computing, Engineering and Digital Technologies, Teesside University, UK (victorchang.research111@gmail.com)

R. Buyya is with Cloud Computing and Distributed Systems (CLOUDS) Laboratory, School of Computing and Information Systems, The University of Melbourne, Australia (e-mail: rbuyya@unimelb.edu.au).

G. Sun is with Key Lab of Optical Fiber Sensing and Communications (Ministry of Education), University of Electronic Science and Technology of China, Chengdu, China; and he is also with Agile and Intelligent Computing Key Laboratory of Sichuan Province, Chengdu, China (e-mail: gangsun@uestc.edu.cn).

requirements, we first formulate the SFC deployment problem with the SLA requirements by using linear programming.

- To satisfy the security requirements of SFC requests, when the security of each service node and each physical link is given, we propose a maximal-security SFCs deployment algorithm (MS).
- To further reduce the deployment cost, we present a minimal-cost and SLA-guaranteed SFC deployment algorithm (MCSG).
- Furthermore, to both reduce the deployment cost and blocking ratio, we propose a minimal-cost and SLA-guaranteed SFCs deployment algorithm with feedback adjustment (MCSG-FA).

The resting of this paper is arranged as follows. Section II introduces related work. Section III models the studied SFC deployment problem. Section IV presents the proposed SFC deployment algorithms. We evaluate our proposed algorithms in Section V. Finally, Section VI concludes this work.

## II. RELATED WORK

### A. VNF deployment in cloud computing

NFV has become a key technology for improving the network flexibility, which is proposed to transfer the traditional network functions to the VNFs. Multiple VNFs consist of SFC in a specific order, and these SFCs are deployed into the cloud network for providing services to users. There are many researches about SFC deployment in cloud computing [10, 18-23].

In [10], the author studied the problem of traffic-aware and energy-efficient SFC deployment and designed a sampling-based Markov approximation and matching-theoretic algorithm (SAMA) to deploy SFCs into the cloud network for minimizing the deployment cost, but the authors did not consider the SFC deployment in fog computing. In [18], the author researched the shared pipeline problem in the environment of NFV. The study transmitted a plurality of data packets by utilizing the shared pipeline for reducing the core computing resources requirement, but it resulted in increasing the length of the pipeline. In order to solve the problem, the authors presented two heuristic algorithms to balance between the core computing resources requirement and the pipeline length for reducing network latency and resource consumption. To reduce the total VNF delay (including the processing delay and the transmission delay), Long Qu et al. [19] studied the optimal scheduling problem of VNFs. They proposed a heuristic algorithm for optimizing the scheduling of VNFs based on the genetic algorithm can lower the total scheduling time by up to 20%.

The authors in [20] studied the overhead caused by virtual network functions and proposed a new framework based on the programmable software and hardware to obtain the flexibility and the high performance of NFV, and also presented a performance-aware VNF deployment algorithm. To ensure the quality of service (QoS) and reduce the energy

consumption of servers, in [21], the authors studied the VNF migration to adjust the workloads of the servers dynamically and presented corresponding heuristic algorithms to reduce the total cost.

In [22], Marcelo Caggiani Luizelli et al. researched the efficient deployment of large-scale VNFs and virtual links, and presented a heuristic algorithm to solve this problem to optimize the virtual network function deployment and reduce resource costs while meeting the network traffic demand. To minimize the placement cost, the authors in [23] researched the optimal placement of the service function chains and put forward a graph algorithm based on a matrix and multi-stage optimization to achieve the goal of reducing costs.

These researches [18-23] studied the problem of deploying SFCs in cloud computing, but they did not consider the NFV security and utilization of the federated environment of the cloud-fog network to provide better services.

### B. NFV and fog computing

More and more users are accessing the cloud network, centralized cloud computing is facing a big challenge. Thus, the distributed fog computing is proposed to extend and supplement the centralized cloud computing to solve these challenges [24-27]. There are many studies about fog computing in recent years [28-34].

In [28], the authors introduced fog computing into the Internet of things. They put forward a fog computing based model for the problem of face identification, for improving the processing efficiency and reducing the network delay. Kai Liang et al. introduced fog computing into the radio access networks, which combining virtualization and SDN to slice the resources of radio access networks to improve the flexibility of radio access networks [29]. In [30], the authors studied a fog computing-based model of Internet access networks by utilizing virtual machines to host the business of fog network for reducing network latency and improving user experience.

The authors in [31] studied the problem of fog computing access control and proposed a channel encryption and decryption model to improve the security of fog computing to defend network attacks. In [32], Seongjin Park et al. studied the vehicular network's connection problem by utilizing fog computing. In this research, they employed fog node to collect information on the mobile vehicle to realize the corresponding vehicle service to achieve quick connection recovery when a failure occurs. The authors in [33] took into account the problem that fog computing resources cannot satisfy the requirements of vehicle users in the peak hour. The fog vehicular computing concept was proposed to balance the needs of vehicle users and achieve a high utilization of fog computing resources.

In [9], the authors discussed the fusion problem of 5G, cloud-fog computing and NFV, and then put forward a fusion and open architecture to provide the continuous management from cloud computing to fog computing. To meet the performance requirements of 5G services, Ricard Vilalta et al.

put forward an NFV architecture for the federated environment of cloud-fog network to provide performance assurance for 5G services [34].

In [28-33], fog computing was widely applied to the Internet of things, radio access network and vehicular network and provided network services to users, but these studies did not research NFV. The research projects [9, 13, 34] studied the NFV problem by utilizing the advantages of fog computing. They only proposed the NFV architecture of the fog computing environment, but the NFV deployment algorithm is not proposed, so deploying NFV in the fog network needs to be studied.

### C. *SSLA in cloud/fog computing*

With the explosive growth of service requests and virus attacks are more frequent, ensuring the SSLA of services has become a big challenge. Therefore, there are many studies about SSLA in network function virtualization [14, 35-38].

In NFV, due to the sharing of underlying resources and the live migration of VNFs, VNFs are vulnerable to the shared resource misuse attack and the side-channel attack. In order to solve these security threats, the authors in [14] first classified the attacks and then proposed corresponding solutions. Since SSLA of VNF is very important, the authors discussed the security of NFV architecture and the influence of the outsider attacks and insider attacks [35]. However, that work did not give a corresponding solution to defend these attacks. Due to network function virtualization may bring network attacks to services, it is necessary to enhance the service security. Thus, research [36] proposed a security framework to ensure SFC security.

In recent years, distributed denial of service (DDoS) attacks continue to increase. At the same time, the traditional defense methods are not strong enough. To solve this problem, in [37], Bahman Rashidi et al. presented the DDoS defense mechanism to achieve a collaboration network based on “domain-helps-domain” to deal with a lot of DDoS attacks. Although the DDoS defense mechanism can effectively handle many DDoS attacks, it may not be able to deal with other attacks effectively.

In order to lower the impact of server failures on the network services, the authors [38] studied the high availability deployment problem of the service function chain. They presented an SFC deployment algorithm based on service backup. The SFC deployment algorithm can improve the survivability of network services, but it cannot guarantee the SSLA of services.

The studies [14, 35-38] only proposed secure architectures of NFV to guarantee the security of the service from the architecture layer, which cannot defend against all attacks and did not consider utilizing the federated environment of the cloud-fog network to provide more secure services for users. In [39-41], the authors studied the security of fog computing and proposed some architectures and defense mechanisms to guarantee fog computing security. Additionally, the fog radio access network has smaller coverage than the cloud network,

which helps deploy defense hardware to defend attacks. Thus, the fog radio access network can provide higher security than the cloud network does. Therefore, the security deployment problem of SFC is worth further research for providing more secure services for users. The authors in [42] studied the security-aware virtual network mapping problem in the cloud environment, but they did not consider the fog computing environment. It is not appropriate to take the least security value of all links in a path as the security value of the path when the authors computed the security of the embedding path in that research.

## III. PROBLEM DESCRIPTION AND MODELLING

### A. *Problem description*

In this research, the physical network and SFC deployment requests of mobile users with SSLA requirements are given. Similar with Ref. [42], we consider each physical node and each physical link with a security possibility that can defend attacks. Under the given security level of each physical node and link, we propose SFC deployment algorithms for reducing the blocking ratio and the deployment cost while meeting the SSLAs of mobile users.

### B. *Physical network*

In this work, the physical network includes two parts: the centralized cloud network and multiple distributed FRANs. An example of physical network is shown in Fig.1. We indicate the physical network as  $G_P = (N_P, E_P)$ . Where,  $N_P = \{n_1, n_2, \dots, n_{|NP|}\}$  indicates the set of the physical nodes in the physical network,  $|NP|$  indicates the number of the physical nodes.  $E_P = \{l_1, l_2, \dots, l_{|EP|}\}$  denotes the set of the physical links.  $|EP|$  indicates the number of physical links.

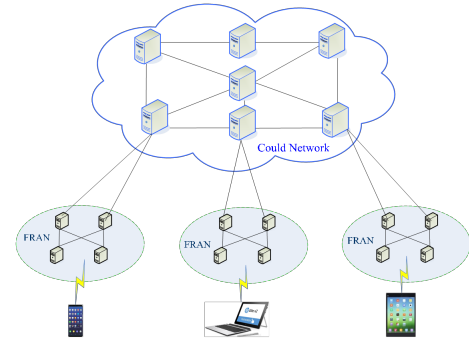


Fig.1 An example of physical network

**Resource constraints of physical network:** we define the physical network resource constraints as  $RC = (C_{NP}, C_{EP}, S_{NP}, S_{EP}, L_{NP})$ .

**Resource attributes of physical node:** we use  $C_{NP}$  to represent the resource attributes set of the physical nodes, which consist of the unit cost  $p(n_i)$  of the resource of physical node and the capacity  $c(n_i)$  of the computing resource of physical node.

**Resource attributes of physical link:**  $C_{EP}$  denotes the resource attributes set of physical links, which consists of the

unit cost  $p(l_i)$  of the resource of physical link and the resource capacity  $b(l_i)$  of physical link.

**Security level of physical node:** It is a numerical concept of the security level of the physical node [42]. We use  $S_{NP} = \{s(n_1), s(n_2), \dots, s(n_{|NP|})\}$  to denote the security level set of physical nodes.

**Security level of physical link:** It's a numerical concept of the security level of physical link [42]. We use  $S_{EP} = \{s(l_1), s(l_2), \dots, s(l_{|EP|})\}$  to denote the security level set of physical link. While maintaining the security of the physical node/link may need to purchase the specialized hardware or invest in human resources, so the higher the security level, the higher the cost of unit resource of the physical node/link.

**Locations of physical nodes:** We use the notation  $L_{NP} = \{L(n_1), L(n_2), \dots, L(n_{|NP|})\}$  to denote the set of locations of all physical nodes.

### C. SFC request

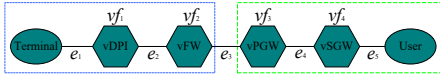


Fig.2 An example of SFC request

An SFC request of mobile user includes two parts: VNFs of cloud network and VNFs of fog radio access network. In traditional access network, the network functions of radio access network are realized by dedicated hardware, which performs the network functions of user's traffic billing, user management and IP address allocation to access to an external network. In order to facilitate the management of users, these network functions are all implemented in access network. In NFV environment, these network functions implemented through virtualization also need to be implemented in the corresponding access network to manage users effectively. Fig.2. shows an example of an SFC request. Similar with the physical network, we can denote an SFC deployment request as  $G_F = (N_F, E_F)$ . Where  $N_F = \{vf_1, vf_2, \dots, vf_{|NF|}\}$  indicates the set of the VNFs,  $|NF|$  is the number of the VNFs in the SFC request.  $E_F = \{e_1, e_2, \dots, e_{|EF|}\}$  denotes the set of links of the SFC request.  $|EF|$  indicates the number of links in the SFC request.

**Deployment constraints:** We define the deployment constraints of the SFC deployment request as  $DC = (C_{NF}, C_{EF}, SR, L_{NF}, L_T, L_U)$ .

**Resource constraints of VNFs:** We define  $C_{NF} = \{\varepsilon(vf_1), \varepsilon(vf_2), \dots, \varepsilon(vf_{|NF|})\}$  as the computing resource constraint set of all VNFs.

**Resource constraints of SFC links:**  $C_{EF} = \{\varepsilon(e_1), \varepsilon(e_2), \dots, \varepsilon(e_{|EF|})\}$  represents the bandwidth resource constraint set of SFC links.

**SSLA requirement of an SFC request:** We define  $SR$  as the overall SSLA constraints of an SFC request.

**Location constraints of VNFs, service terminal and mobile user:** We use  $LC_{NF} = \{LC(vf_1), LC(vf_2), \dots, LC(vf_{|NF|})\}$  to denote the set of location constraints of all VNFs. VNFs of the cloud network can only be deployed into cloud network, and VNFs of the fog radio access network can only be mapped into FRAN in which the mobile user is located.  $L_T$

represents the location of service terminal.  $L_U$  denotes the location of mobile user.

### D. Modelling for SFC deployment

For provisioning an SFC request, we have to effectively deploying the VNFs and links. The deployment for VNFs can be formulated as:

$$DS : (N_F, C_{NF}) \xrightarrow{DS} (N_{P1}, C_{NP1}),$$

$$DS(vf_i) \in N_{P1}, \forall vf_i \in N_F,$$

$$A(DS(vf_i)) \geq \varepsilon(vf_i), \forall vf_i \in N_F,$$

$$Z(LC(vf_i), q) \in \{0, 1\}, \forall vf_i \in N_F, \forall q \in \{0, 1, \dots, Q\},$$

$$L(DS(vf_i)) \in \{0, 1, 2, \dots, Y\}, \forall vf_i \in N_F,$$

$$Z(LC(vf_i), L(DS(vf_i))) = 1, \forall vf_i \in N_F,$$

where  $DS = (DS_N, DS_E)$  denotes the deployment solution of the SFC request;  $DS_N = \{DS(vf_1), DS(vf_2), \dots, DS(vf_{|NF|})\}$  denotes the set of placement solutions of all VNFs;  $DS_E = \{DS(e_1), DS(e_2), \dots, DS(e_{|EF|})\}$  denotes the set of placement solutions of all SFC links.  $N_{P1} \subset N_P$  indicates a subset of the physical nodes for hosting the VNFs;  $C_{NP1} \subset C_{NP}$  describes the node resources allocated to the SFC request;  $DS(vf_i)$  denotes a physical node for hosting the  $i$ -th VNF  $vf_i$ ;  $A(DS(vf_i))$  denotes the available node resources of the physical node  $DS(vf_i)$ ;  $q \in \{0, 1, \dots, Q\}$  indicates the number of the network areas;  $Z(LC(vf_i), q) \in \{0, 1\}$  indicates a binary variable, when  $Z(LC(vf_i), q) = 1$  denotes that VNF  $vf_i$  can be deployed into the network area, otherwise  $Z(LC(vf_i), q) = 0$ .  $L(DS(vf_i))$  describes the number of network areas of physical node  $DS(vf_i)$ .  $Z(LC(vf_i), L(DS(vf_i))) = 1$  describes that physical node  $DS(vf_i)$  satisfies the location constraint of VNF  $vf_i$ , otherwise  $Z(LC(vf_i), L(DS(vf_i))) = 0$ .

In this work, we deploy the SFC links when we deploy the VNFs, and we formulate the deployment of SFC links as:

$$DS : (E_F, C_{EF}) \xrightarrow{DS} (P1, C_{EP1}),$$

$$DS(e_i) = p_{e_i}, \forall e_i \in E_F, \exists p_{e_i} \in P1,$$

$$B(p_{e_i}) = \min_{l_j \in p_{e_i}} \{b(l_j)\} \geq \varepsilon(e_i), \exists p_{e_i} \in P1,$$

wherein  $P1$  indicates a subset of the physical paths;  $C_{EP1}$  denotes the resources of physical links allocated for the SFC request;  $p_{e_i}$  and  $DS(e_i)$  show a physical path for hosting the SFC link  $e_i$ ;  $B(p_{e_i})$  used to denote the available bandwidth resource of physical path  $p_{e_i}$ .

$TSecurity(DS)$

$$= \left\{ \prod_{i=1}^{|NF|} VNFSecurity(DS(vf_i)) \right\} \left\{ \prod_{i=1}^{|EF|} PathSecurity(DS(e_i)) \right\} \quad (1)$$

$$= \left\{ \prod_{i=1}^{|NF|} s(DS(vf_i)) \right\} \left\{ \prod_{i=1}^{|EF|} \prod_{l_k \in DS(e_i)} s(l_k) \right\}$$

For a given deployment solution  $DS$  of an SFC, the security can be computed through Equation (1).

To satisfy the SSLA requirements of an SFC request, we first consider maximizing the security of the SFC and model the maximal-security deployment as the following linear programming problem (2).

$$\max \left( \left\{ \prod_{i=1}^{|NF|} S(DS(vf_i)) \right\} \left\{ \prod_{i=1}^{|EF|} \prod_{l_k \in DS(e_i)} s(l_k) \right\} \right)$$

s. t.

$$A(DS(vf_i)) \geq \varepsilon(vf_i), \forall vf_i \in N_F$$

$$Z(LC(vf_i), q) \in \{0, 1\}, \forall vf_i \in N_F, \forall q \in \{0, 1, \dots, Q\} \quad (2)$$

$$L(DS(vf_i)) \in \{0, 1, 2, \dots, Q\}, \forall vf_i \in N_F$$

$$Z(LC(vf_i), L(DS(vf_i))) = 1, \forall vf_i \in N_F$$

$$DS(e_i) = p_{e_i}, \quad \forall e_i \in E_F, \exists p_{e_i} \in P1$$

$$B(p_{e_i}) = \min_{l_j \in p_{e_i}} \{b(l_j)\} \geq \varepsilon(e_i), \quad \exists p_{e_i} \in P1$$

Suppose the security of the SFC deployment solved by the linear programming (2) does not meet the SSLA requirement of the SFC request. In that case, the SFC request will be rejected because the current network cannot provide a deployment with sufficient security. Since the security level is higher, the unit cost of the physical node/link is the higher, and the maximal-security implementation may lead to a higher deployment cost. To minimize the total deployment cost when the SSLA requirements of the SFC deployment request is guaranteed, we model the minimal-cost and SSLA-guaranteed SFC provision as the linear programming problem (3).

$$\min \left( \sum_{vf_i \in N_F} P(DS(vf_i))\varepsilon(vf_i) + \sum_{e_i \in E_F} \sum_{l_j \in p_{e_i}} P(l_j)\varepsilon(e_i) \right)$$

s. t.

$$\left\{ \prod_{i=1}^{|NF|} S(DS(vf_i)) \right\} \left\{ \prod_{i=1}^{|EF|} \prod_{l_k \in DS(e_i)} s(l_k) \right\} \geq SR$$

$$A(DS(vf_i)) \geq \varepsilon(vf_i), \forall vf_i \in N_F$$

$$Z(LC(vf_i), q) \in \{0, 1\}, \forall vf_i \in N_F, \forall q \in \{0, 1, \dots, Q\} \quad (3)$$

$$L(DS(vf_i)) \in \{0, 1, 2, \dots, Q\}, \forall vf_i \in N_F$$

$$Z(LC(vf_i), L(DS(vf_i))) = 1, \forall vf_i \in N_F$$

$$DS(e_i) = p_{e_i}, \quad \forall e_i \in E_F, \exists p_{e_i} \in P1$$

$$B(p_{e_i}) = \min_{l_j \in p_{e_i}} \{b(l_j)\} \geq \varepsilon(e_i), \quad \exists p_{e_i} \in P1$$

## IV. ALGORITHM DESIGN

Due to the SFC deployment with the SSLA requirement is an NP-hard problem, the linear programming cannot gain a deployment solution  $DS$  in polynomial time. To solve this problem, we put forward a security-aware SFC deployment (SASFCD) algorithm. The SASFCD algorithm can be used in three different scenarios by calling different sub-algorithms, i.e., maximal-security SFC deployment algorithm (MS), minimal-cost and SSLA-guaranteed SFCs deployment algorithm (MCSG) and minimal-cost and SSLA-guaranteed SFCs deployment algorithm with feedback adjustment (MCSG-FA). We assume that the SFC deployment requests follow the Poisson process to arrive dynamically. All arrived SFC requests are stored in the queue *ArrivalsFC*. The finished SFC requests are stored in the set *FinishedSFC*. Each SFC deployment request in the queue *ArrivalsFC* is deployed one by one. We use  $SFC_{blo}$  to indicate the set of blocked SFC requests owing to the resource limitation. *Algorithm 1* shows the SASFCD algorithm.

---

### **Algorithm 1:** SASFCD algorithm

---

**Input:** 1. Physical network  $G_P = (N_P, E_P)$  and resource constraints  $RC = (C_{NP}, C_{EP}, S_{NP}, S_{EP}, LC_{NP})$ .

2. SFC request queue *ArrivalsFC*.

**Output:** Total deployment cost  $TCost$  and the blocked SFC set  $SFC_{blo}$ .

- 1: Initialization: set  $TCost=0$  and  $SFC_{blo}=\emptyset$ ;
  - 2: **while** *ArrivalsFC*  $\neq \emptyset$ , **do**
  - 3:   Updating resources according to the set *FinishedSFC*.
  - 4:   Call MS, MCSG or MCSG-FA for deploying the first SFC request  $SFC_1$  in *ArrivalsFC*.
  - 5:   **if** the deployment solution  $DS$  for  $SFC_1$ ,  $DS \neq \emptyset$ , **then**
  - 6:     updating  $TCost$  and the physical network.
  - 7:   **else**
  - 8:      $SFC_{blo} = SFC_{blo} \cup \{SFC_1\}$ .
  - 9:   **end if**
  - 10:    $ArrivalsFC = ArrivalsFC \setminus \{SFC_1\}$ .
  - 11: **end while**
  - 12: **return**  $TCost$  and  $SFC_{blo}$ .
- 

The MS algorithm is used to maximize the security of the SFC deployment solution. In the MS algorithm, we use the maximal-security strategy as a guide strategy for deploying VNF into the most secure physical node and finding the most secure path to maximize the security of the placement solution of SFC, thus get the most secure deployment solution. When we deploy VNF  $vf_i$  into the physical node  $n_j$ , and find the most secure path  $p_{e_i}$ , we can find the most secure path  $p^{i+1}(n_j, L_U)$  from the current physical node  $n_j$  to the user. The aim is to optimize the deployment solution and improve the blocking ratio to ensure the security of the deployment solution. If the security of the SFC deployment solved by the MS algorithm cannot meet the SSLA requirement of the SFC request, the SFC deployment request will be rejected.

The security of VNF  $vf_i$  deployed into the physical node  $n_j$ ,  $VNFSecurity(vf_i \rightarrow n_j)$ , that is defined in Equation (4).

$$VNFSecurity(vf_i \rightarrow n_j) = s(n_j) \quad (4)$$

The deployment cost of VNF  $vf_i$  deployed into the physical node  $n_j$ ,  $VNFCost(vf_i \rightarrow n_j)$ , that is defined as in Equation (5).

$$VNFCost(vf_i \rightarrow n_j) = \varepsilon(vf_i)p(n_j) \quad (5)$$

The security of the physical path  $p_{ei}$  for hosting SFC link  $e_i$ ,  $PathSecurity(p_{ei})$ , is denoted as in Equation (6).

$$PathSecurity(p_{ei}) = \prod_{l_k \in p_{ei}} s(l_k) \quad (6)$$

The security of the physical path  $p^{i+1}(n_j, L_U)$  for hosting link  $(n_j, L_U)$ ,  $PathSecurity(p^{i+1}(n_j, L_U))$ , is denoted as in Equation (7). The physical path  $p^{i+1}(n_j, L_U)$  must meet the link resource requirements of SFC link  $e_{i+1}$ .

$$PathSecurity(p^{i+1}(n_j, L_U)) = \prod_{l_k \in p^{i+1}(n_j, L_U)} s(l_k) \quad (7)$$

The deployment cost of physical path  $p_{ei}$  for hosting SFC link  $e_i$ ,  $PathCost(p_{ei})$ , can be computed in Equation (8).

$$PathCost(p_{ei}) = \sum_{l_k \in p_{ei}} p(l_k)\varepsilon(e_i) \quad (8)$$

The security of VNF  $vf_i$  deployed into the physical node  $n_j$ ,  $TSecurity(vf_i \rightarrow n_j)$ , that is represented as in Equation (9).

$$\begin{aligned} TSecurity(vf_i \rightarrow n_j) &= VNFSecurity(vf_i \rightarrow n_j) \times PathSecurity(p_{ei}) \\ &\times PathSecurity(p^{i+1}(n_j, L_U)) \\ &= s(n_j) \left\{ \prod_{l_k \in p_{ei}} s(l_k) \right\} \left\{ \prod_{l_k \in p^{i+1}(n_j, L_U)} s(l_k) \right\} \end{aligned} \quad (9)$$

The deployment cost of VNF  $vf_i$  deployed into the physical node  $n_j$ ,  $TCost(vf_i \rightarrow n_j)$ , that is denoted as in Equation (10).

$$\begin{aligned} TCost(vf_i \rightarrow n_j) &= VNFCost(vf_i \rightarrow n_j) + PathCost(p_{ei}) \\ &= \varepsilon(vf_i)p(n_j) + \sum_{l_k \in p_{ei}} \{p(l_k)\varepsilon(e_i)\} \end{aligned} \quad (10)$$

The security of the current deployment solution  $DS$  in the MS algorithm,  $TSecurity(DS)'$ , is defined as in Equation (11).

$$\begin{aligned} TSecurity(DS)' &= PathSecurity(p^{i+1}(n_j, L_U)) \\ &\times \prod_{k=1}^i \{VNFSecurity(DS(vf_k)) \times PathSecurity(DS(e_i))\} \\ &= \left\{ \prod_{l_k \in p^{i+1}(n_j, L_U)} s(l_k) \right\} \left\{ \prod_{k=1}^i [s(DS(vf_k)) \prod_{l_k \in DS(e_i)} s(l_k)] \right\} \end{aligned} \quad (11)$$

The total deployment cost of the current deployment solution  $DS$ ,  $TCost(DS)$ , can be defined as in Equation (12).

$TCost(DS)$

$$\begin{aligned} &= \sum_{i=1}^{|NF|} VNFCost(DS(vf_i)) + \sum_{i=1}^{|EF|} PathCost(DS(e_i)) \\ &= \sum_{i=1}^{|NF|} \varepsilon(vf_i)p(DS(vf_i)) + \sum_{i=1}^{|EF|} \sum_{l_k \in DS(e_i)} p(l_k)\varepsilon(e_i) \end{aligned} \quad (12)$$

---

### Algorithm 2: MS algorithm

---

**Input:** 1. Physical network  $G_P = (N_P, E_P)$  and resource constraints  $RC = (C_{NP}, C_{EP}, S_{NP}, S_{EP}, L_{NP})$ .  
2. SFC request  $G_F = (N_F, E_F)$  and deployment constraints  $DC = (C_{NF}, C_{EF}, SR, LC_{NF}, L_T, L_U)$ .

**Output:** Deployment solution  $DS$  and total deployment cost  $TCost(DS)$ .

- 1: **for** each VNF  $vf_i, i=1,2,\dots, |NF|, vf_i \in N_F$ , **do**
  - 2:   **for** each physical node  $n_j \in N_P$ , **do**
  - 3:     **if** the node  $n_j$  meets the location constraint of  $vf_i$ , **then**
  - 4:       Try to deploy  $vf_i$  into the physical node  $n_j$ , calculate the security of the deployment solution of  $vf_i$   $VNFSecurity(vf_i \rightarrow n_j)$  and the deployment cost  $VNFCost(vf_i \rightarrow n_j)$  according to Eqs. (4) and (5).
  - 5:       Find the secure paths  $p_{ei}$  and  $p^{i+1}(n_j, L_U)$ ; Calculate the security of the path  $p_{ei}$   $PathSecurity(p_{ei})$ , the security of the path  $p^{i+1}(n_j, L_U)$   $PathSecurity(p^{i+1}(n_j, L_U))$  and the deployment cost  $PathCost(p_{ei})$  according to Eqs. (6), (7) and (8); Calculate the security  $TSecurity(vf_i \rightarrow n_j)$  and the total deployment cost  $TCost(vf_i \rightarrow n_j)$  according to Equations (9) and (10).
  - 6:     **end if**
  - 7:   **end for**
  - 8:   Find the deployment solution of VNF  $vf_i$  with the maximal security  $TSecurity(vf_i \rightarrow n_j)$ , and store the deployment solutions of  $vf_i$  and  $e_i$  in  $DS$ .
  - 9:   **if** the deployment solution cannot be found, **then**
  - 10:     Clear  $DS$  and let  $TCost(DS) = 0$ .
  - 11:   **return**  $DS$  and  $TCost(DS)$ .
  - 12: **end if**
  - 13:   Calculate the security  $TSecurity(DS)'$  according to Equation (11).
  - 14:   **if**  $TSecurity(DS)' < SR$ , **then**
  - 15:     Clear  $DS$  and let  $TCost(DS) = 0$ .
  - 16:   **return**  $DS$  and  $TCost(DS)$ .
  - 17: **end if**
  - 18: **end for**
  - 19: Find the most secure path  $p_{e_{|EF|}}$  and store it in  $DS$ , calculate the total deployment cost  $TCost(DS)$  according to Equation (12).
  - 20: **return**  $DS$  and  $TCost(DS)$ .
- 

A higher security of the deployment leads to a higher total deployment cost in the MS algorithm. To minimize the total deployment cost, while satisfying the SSLA requirement of the SFC request, we propose the MCSG algorithm. In MCSG algorithm, we use the minimal-cost and SSLA-guaranteed

strategy to deploy VNFs for minimizing the total placement cost of the SFC deployment solution. In MCSG algorithm, similar to the MS algorithm, we will find the most secure path  $p^{i+1}(n_j, L_U)$  from the current physical node  $n_j$  to user, to reduce the total cost.

$MaxSecurity(p^{i+1}(n_j, L_U))$  denotes the maximal security of the pre-deployment solution of the rest of VNFs that we pre-deploy the rest of VNFs into the most secure physical node on the physical path  $p^{i+1}(n_j, L_U)$ , it is defined as in Equation (13). Where,  $n_t \in p^{i+1}(n_j, L_U)$  denotes the physical node on the physical path  $p^{i+1}(n_j, L_U)$ .

$$MaxSecurity(p^{i+1}(n_j, L_U)) = \prod_{k=i+1}^{|NF|} \max \{s(n_t), \forall n_t \in p^{i+1}(n_j, L_U)\} \quad (13)$$

The deployment cost of physical path  $p^{i+1}(n_j, L_U)$  for hosting link  $(n_j, L_U)$ ,  $PathCost(p^{i+1}(n_j, L_U))$ , can be computed in Equation (14).

$$PathCost(p^{i+1}(n_j, L_U)) = \sum_{l_k \in p^{i+1}(n_j, L_U)} p(l_k) \varepsilon(e_{i+1}) \quad (14)$$

The security of the current deployment solution  $DS$  in the MCSG algorithm,  $TSecurity(DS)''$ , is denoted as in Eq. (15).

$$\begin{aligned} TSecurity(DS)'' &= PathSecurity(p^{i+1}(n_j, L_U)) \\ &\times MaxSecurity(p^{i+1}(n_j, L_U)) \\ &\times \prod_{k=1}^i \{VNFSecurity(DS(vf_k)) \times PathSecurity(DS(e_i))\} \quad (15) \\ &= \left\{ \prod_{l_k \in p^{i+1}(n_j, L_U)} s(l_k) \right\} \left\{ \prod_{k=i+1}^{|NF|} \max \{s(n_t), \forall n_t \in p^{i+1}(n_j, L_U)\} \right\} \\ &\left\{ \prod_{k=1}^i [s(DS(vf_k))] \prod_{l_k \in DS(e_i)} s(l_k) \right\} \end{aligned}$$

The total deployment cost of VNF  $vf_i$  deployed into the physical node  $n_j$  in the MCSG or MCSG-FA algorithm,  $TCost(vf_i \rightarrow n_j)''$ , that is denoted as in Equation (16).

$$\begin{aligned} TCost(vf_i \rightarrow n_j)'' &= VNFcost(vf_i \rightarrow n_j) + PathCost(p_{e_i}) \\ &+ PathCost(p^{i+1}(n_j, L_U)) \quad (16) \\ &= \varepsilon(vf_i) p(n_j) + \sum_{l_k \in p_{e_i}} p(l_k) \varepsilon(e_i) + \sum_{l_k \in p^{i+1}(n_j, L_U)} p(l_k) \varepsilon(e_{i+1}) \end{aligned}$$

---

### Algorithm 3: MCSG algorithm

---

**Input:** 1. Physical network  $G_P = (N_P, E_P)$  and resource constraints  $RC = (C_{NP}, C_{EP}, S_{NP}, S_{EP}, L_{NP})$ .

2. SFC request  $G_F = (N_F, E_F)$  and deployment constraints  $DC = (C_{NF}, C_{EF}, SR, LC_{NF}, LT, LU)$ .

**Output:** Deployment solution  $DS$  and total deployment cost  $TCost(DS)$ .

- 1: **for** each VNF  $vf_i, i=1,2,\dots, |NF|, vf_i \in N_S$ , **do**
- 2:   **for** each physical node  $n_j \in N_P$ , **do**
- 3:     **if** the node  $n_j$  meets the location constraint of  $vf_i$ , **then**
- 4:       Try to deploy  $vf_i$  into the physical node  $n_j$ ; Calculate

the security of the deployment solution of  $vf_i$   $VNFSecurity(vf_i \rightarrow n_j)$  and the deployment cost  $VNFcost(vf_i \rightarrow n_j)$  according to Eqs. (4) and (5).

- 5:     Find the most secure paths  $p_{e_i}$  and  $p^{i+1}(n_j, L_U)$ ; Calculate the security of path  $p_{e_i}$   $PathSecurity(p_{e_i})$ , the security of path  $p^{i+1}(n_j, L_U)$   $PathSecurity(p^{i+1}(n_j, L_U))$ , the maximal security of the pre-deployment of the rest of VNFs  $MaxSecurity(p^{i+1}(n_j, L_U))$ , the deployment cost  $PathCost(p_{e_i})$  and the deployment cost  $PathCost(p^{i+1}(n_j, L_U))$  according to Equations (6), (7), (13), (8) and (14), respectively; Calculate the security  $TSecurity(DS)''$  and deployment cost  $TCost(vf_i \rightarrow n_j)''$  according to Eqs. (15) and (16).
  - 6:     **end if**
  - 7:   **end for**
  - 8: Find the deployment solution of VNF  $f_i$  with the minimal deployment cost  $TCost(vf_i \rightarrow n_j)''$  and the total security  $TSecurity(DS)'' \geq SR$ ; Store the deployment solutions of  $vf_i$  and  $e_i$  in  $DS$ .
  - 9:   **if** the deployment solution cannot be found, **then**
  - 10:     Clear  $DS$  and let  $TCost(DS) = 0$ .
  - 11:   **return**  $DS$  and  $TCost(DS)$ .
  - 12:   **end if**
  - 13: **end for**
  - 14: Find the most secure path  $p_{e_i|EF|}$  and store it in  $DS$ ; Calculate the deployment cost  $TCost(DS)$  according to Equation (12).
  - 15: **return**  $DS$  and  $TCost(DS)$ .
- 

MCSG algorithm uses minimal-cost and SLA-guaranteed strategy for deploying VNF to minimize the total placement cost. However, MCSG algorithm reduces the total placement cost by using the security-guaranteed strategy that will cause an increase in the blocking ratio. Therefore, we propose the MCSG-FA algorithm to improve the blocking ratio. In the MCSG-FA algorithm, we first call the MS algorithm to get an initial deployment solution with maximal security. We try to find a deployment solution with minimal total deployment cost. If we find a new deployment solution, and the security of the new deployment solution meets the SLA requirement of the SFC requests. The total placement cost of the new placement solution is less than that of the initial deployment solution, and we use the new placement solution to replace the initial one. The MCSG-FA algorithm can improve the total deployment cost and the blocking ratio through the feedback adjustment approach.

The security of current deployment solution  $DS'$  in the MCSG-FA algorithm,  $TSecurity(DS')$ , can be computed by Equation (17).

$$\begin{aligned}
& TSecurity(DS') \\
&= \prod_{k=1}^i \{VNFSecurity(DS'(vf_k)) \times PathSecurity(DS'(e_i))\} \quad (17) \\
&= \left\{ \prod_{k=1}^i [s(DS'(vf_k)) \prod_{l_k \in DS'(e_i)} s(l_k)] \right\}
\end{aligned}$$

The total deployment cost of current deployment solution  $DS'$  in the MCSG-FA algorithm,  $TCost(DS')$ , can be defined in Equation (18).

$$\begin{aligned}
& TCost(DS') \\
&= \sum_{i=1}^{|NF|} VNFcost(DS'(vf_i)) + \sum_{i=1}^{|EF|} PathCost(DS'(e_i)) \quad (18) \\
&= \sum_{i=1}^{|NF|} \varepsilon(vf_i) p(DS'(vf_i)) + \sum_{i=1}^{|EF|} \sum_{l_k \in DS'(e_i)} p(l_k) \varepsilon(e_i)
\end{aligned}$$

---



---

#### Algorithm 4: MCSG-FA algorithm

---

**Input:** 1. Physical network  $G_P = (N_P, E_P)$  and resource constraints  $RC = (C_{NP}, C_{EP}, S_{NP}, S_{EP}, L_{NP})$ .  
2. SFC request  $G_F = (N_F, E_F)$  and deployment constraints  $DC = (C_{NF}, C_{EF}, SR, LC_{NF}, L_T, L_U)$ .

**Output:** Deployment solution  $DS$  and total deployment cost  $TCost(DS)$ .

```

1: Call MS algorithm to achieve  $DS$  and  $TCost(DS)$ .
2: if  $DS \neq \emptyset$ , then
3:   for each VNF  $vf_i, i=1,2,\dots, |NF|, vf_i \in N_F$ , do
4:     for each physical node  $n_j \in N_P$ , do
5:       if node  $n_j$  meets the location constraint of  $vf_i$ , then
6:         Try to deploy  $vf_i$  into the physical node  $n_j$ ;
         Calculate the security of deployment solution of
          $vf_i$   $VNFSecurity(vf_i \rightarrow n_j)$  and the deployment cost
          $VNFcost(vf_i \rightarrow n_j)$  according to Eqs. (4) and (5).
7:         Find the minimal cost paths  $p_{ei}, p^{i+1}(n_j, L_U)$ ;
         Calculate security  $PathSecurity(p_{ei})$ , deployment
         costs  $PathCost(p_{ei})$  and  $PathCost(p^{i+1}(n_j, L_U))$ ,
         and total deployment cost  $TCost(vf_i \rightarrow n_j)$ 
         according to Eqs. (6), (8), (14) and (16).
8:       end if
9:     end for
10:    Find the deployment solution of VNF  $vf_i$  with the
    minimal deployment cost  $TCost(vf_i \rightarrow n_j)$ , and store the
    deployment solutions of  $vf_i$  and  $e_i$  in  $DS'$ .
11:   if the deployment solution cannot be found, then
12:     Clear  $DS'$  and let  $TCost(DS') = 0$ .
13:   return  $DS$  and  $TCost(DS)$ .
14:   end if
15:   Calculate the security  $TSecurity(DS')$  according to
   Equation (17).
16:   if  $TSecurity(DS') < SR$ , then
17:     Clear  $DS'$  and let  $TCost(DS') = 0$ .
18:   return  $DS$  and  $TCost(DS)$ .
19:   end if
20: end for

```

```

21: Find the minimal-cost path  $p_{e_i|EF}$  and store it in  $DS'$ ;
    Calculate the security  $TSecurity(DS')$  according to
    Equation (17); Calculate the total deployment cost
     $TCost(DS')$  according to Equation (18).

```

```

22: end if

```

```

23: if find a complete solution  $DS'$  and  $TSecurity(DS') \geq SR$ 
    and  $TCost(DS') < TCost(DS)$ , then

```

```

24:   Let  $DS = DS'$ ,  $TCost(DS) = TCost(DS')$ .

```

```

25: end if

```

```

26: return  $DS$  and  $TCost(DS)$ .

```

---

Next, we analyze the complexity and security of these proposed algorithms. In MS algorithm, we use the Dijkstra algorithm to find the shortest path, the complexity of the Dijkstra is  $O(|NP|^2)$ .  $|NP|$  denotes the number of physical servers;  $|NF|$  denotes the number of VNFs in SFC;  $c_1, c_2, c_3, c_4, c_5$  and  $c_6$  are constants. So, the complexity of the MS algorithm can be evaluated as follows. In line 5: we use the Dijkstra algorithm to find the shortest path, so the complexity is  $c_1|NP|^2$ . In line 1~13: the complexity is  $c_2|NF||NP||NP|^2 = c_2|NF||NP|^3$ . In line 19: we use the Dijkstra algorithm to find the shortest path, so the complexity is  $c_3|NP|^2$ . In line 1~20: the complexity is  $c_2|NF||NP|^3 + c_3|NP|^2 = O(|NF||NP|^3)$ . Thus, the complexity of MS algorithm is  $O(|NF||NP|^3)$ .

The complexity of MCSG-FA algorithm can be evaluated as follows. In line 1: we call MS algorithm, the time complexity is  $O(|NF||NP|^3)$ . In line 7: we employ the Dijkstra algorithm to find the shortest path, so the time complexity is  $c_4|NP|^2$ . In line 3~9: the time complexity is  $c_5|NF||NP||NP|^2 = c_5|NF||NP|^3$ . In line 21: we use the Dijkstra algorithm to find the shortest path, so the complexity is  $c_6|NP|^2$ . In line 2~26: the complexity is  $c_5|NF||NP|^3 + c_6|NP|^2 = O(|NF||NP|^3)$ . Thus, the complexity of the MCSG-FA algorithm is  $O(|NF||NP|^3)$ .

Finally, we analyze the security of these algorithms. First, virtualization technology can provide a strong isolation for VNFs to avoid denial of service caused by interference of other VNFs. The strong isolation includes resource isolation, performance isolation, and security isolation, so that each VNF or each virtual link has independence in terms of resources, performance, and security. Furthermore, virtualization technology offers inter- and intra-network QoS provisioning by using a consistent resource controller. Second, we defined the overall security of an SFC deployment in Equation (1) and assumed each physical node and each physical link to have a security level for defending attacks [42]. Furthermore, in our proposed algorithms, when we deploy an SFC, the MS algorithm uses the maximal-security strategy as a guide strategy for deploying VNFs into the most secure physical node and finding the most reliable paths to ensure the security of SFC deployment. The MCSG algorithm uses the minimal-cost and SLA-guaranteed strategies for deploying VNFs to ensure the security of the SFC deployment. The MCSG-FA algorithm first calls the MS algorithm to get an initial deployment, then it tries to find a new deployment to



minimize the total deployment cost while meeting security requirements. Therefore, we can guarantee the security of an SFC request from two aspects of virtualization technology and the deployment strategy in our proposed algorithms.

## V. PERFORMANCE EVALUATION AND ANALYSIS

### A. Simulation environment

In this work, we consider utilizing the federated environment of the cloud-fog network to provide more secure services for more mobile users. Therefore, the physical network is comprised of cloud network (the USANET network, as shown in Fig.3) and multiple FRANs (as shown in Fig. 4). In our simulations, there are 15 fog radio access networks that connect to the black nodes numbered 0, 5, 7, 12, 14, 16, 20, 23, 25, 29, 32, 34, 36, 42 and 44.

We presume that the unit cost of the node resource of each physical node in physical network is  $\log(1/(1-s(n_i)))$ , the unit cost of the resource of each physical link in physical network is  $\log(1/(1-s(l_i)))$ . In our simulations, when we evaluate the total deployment costs and the running time of our proposed algorithms, we assume that the resource capacity constraints of the physical node follow a uniform distribution  $U(50, 80)$ . The resource capacity constraints of the physical link follow a uniform distribution  $U(30, 50)$ . When we evaluate the blocking ratios of all algorithms, we assume that the resource capacity of the physical network is unlimited. Without losing generality, we make assumption that: *i*) Per unit cost of computing resource is 1 unit and per unit cost of bandwidth resources is 1 unit; *ii*) The transmission delay of each core network link is 1 unit.

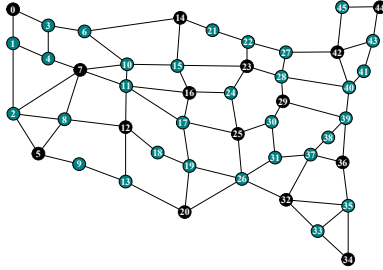


Fig.3 USANET network

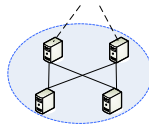


Fig.4 The topology of a FRAN

In our simulations, we assume that 10000 SFC requests arrive dynamically following a Poisson process when the lengths of SFC requests (i.e.,  $n$ ) varies among 5, 6, 7 and 8, respectively. The resource requirements of VNF and SFC link obey a uniform distribution  $U(5, 10)$ . We suppose that the location of the service terminal randomly distributed in a physical network node of cloud network, and the location of the mobile user randomly distributed in a physical network node of FRAN. We first find a most secure path  $p(L_T, L_U)$  from the service terminal to mobile user. Where,  $n_i \in p(L_T, L_U)$

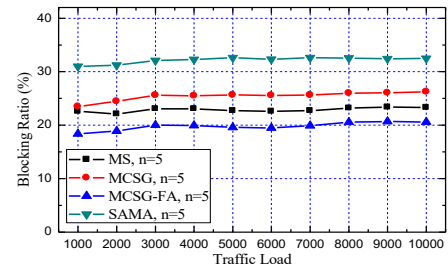
denotes the physical node on the physical path  $p(L_T, L_U)$ . Then, we set the SSLA requirement of an SFC deployment request according to Equation (19).

$$\begin{aligned}
 SR &= PathSecurity(p(L_T, L_U)) \\
 &\times \prod_{i=1}^{[NF]} \left\{ \begin{aligned} &Average\{s(n_i), \forall n_i \in p(L_T, L_U)\} * 2/3 \\ &+ \min\{s(n_i), \forall n_i \in p(L_T, L_U)\} / 3 \end{aligned} \right\} \\
 &= \left\{ \prod_{l_k \in p(L_T, L_U)} s(l_k) \right\} \\
 &\times \prod_{i=1}^{[NF]} \left\{ \begin{aligned} &Average\{s(n_i), \forall n_i \in p(L_T, L_U)\} * 2/3 \\ &+ \min\{s(n_i), \forall n_i \in p(L_T, L_U)\} / 3 \end{aligned} \right\}
 \end{aligned} \tag{19}$$

In our simulations, we will compare our algorithms with the SAMA algorithm, which is presented in [10] for minimizing the total placement cost.

### B. Simulation results and analysis

Fig.5 represents the blocking ratios of the MS, MCSG, MCSG-FA and SAMA algorithms, wherein the length of the SFC request (i.e.,  $n$ ) is changed among 5, 6, 7 and 8. From the results, it can be seen that the blocking ratios of our three algorithms are better than that of SAMA algorithm. The SAMA algorithm is proposed to minimize the total deployment cost, so it does not consider the SSLA requirement of SFC request when looking for a deployment solution. Thus, the blocking ratio of the SAMA algorithm is high. Besides, in our three algorithms, when we deploy VNF  $v_i$  into the physical node  $n_j$  and find the most secure path  $p_{ei}$ , we will find the most secure path  $p^{i+1}(n_j, L_U)$  from the current physical node  $n_j$  to the user to improve the blocking ratio. The MS algorithm uses the maximal-security strategy to deploy VNF into the most secure physical node and find the most reliable paths to maximize the security of the placement solution of SFC. It can maximize the security of the deployment of SFC and guarantee the success ratio. MCSG algorithm uses the minimal-cost and SSLA-guaranteed strategy for deploying VNF to minimize the total placement cost, but lead an increase in the blocking ratio. So, the blocking ratio of the MCSG algorithm is higher than the blocking ratio of MS algorithm. The MCSG-FA algorithm first calls the MS algorithm to get an initial deployment solution, so that the MCSG-FA algorithm has a similar success ratio to the MS algorithm. Then it tries to find a deployment solution with the minimal deployment cost to replace the initial deployment solution. This can reduce the consumption of network resources to improve the blocking ratio further. Therefore, MCSG-FA algorithm has the lowest blocking ratio.



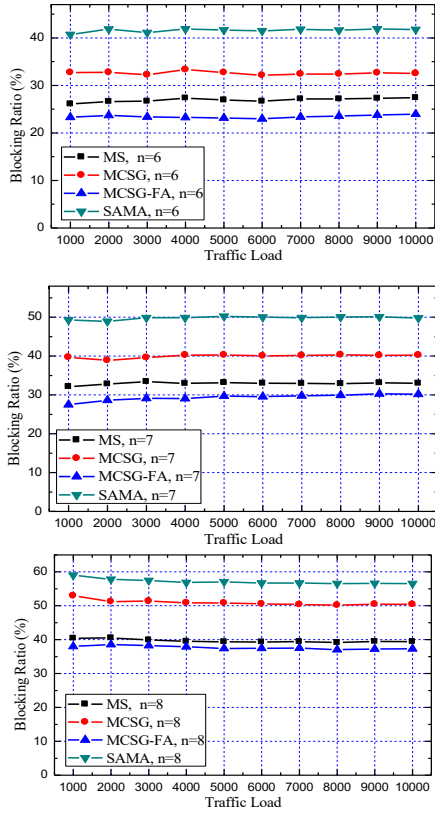


Fig.5 Simulation results of blocking ratio

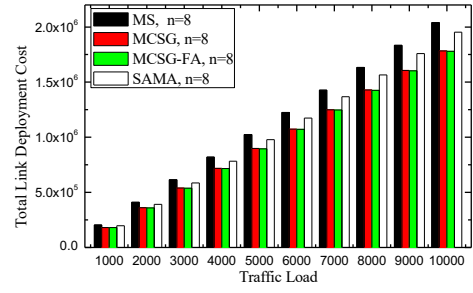
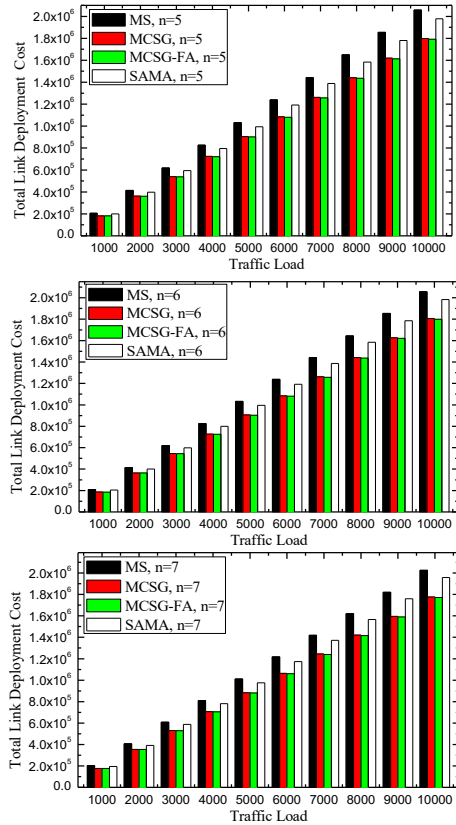


Fig.6 Simulation results of total link deployment cost

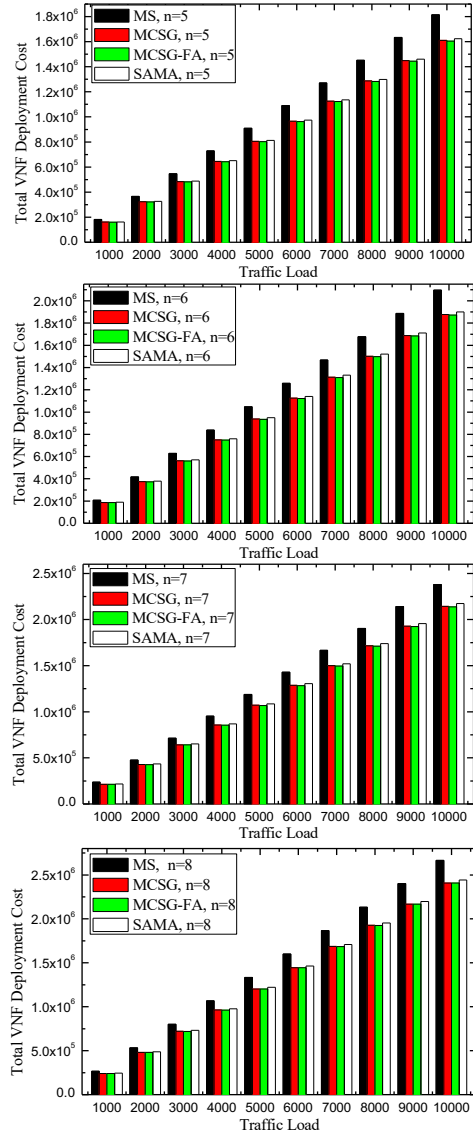


Fig.7 Simulation results of total VNF deployment cost

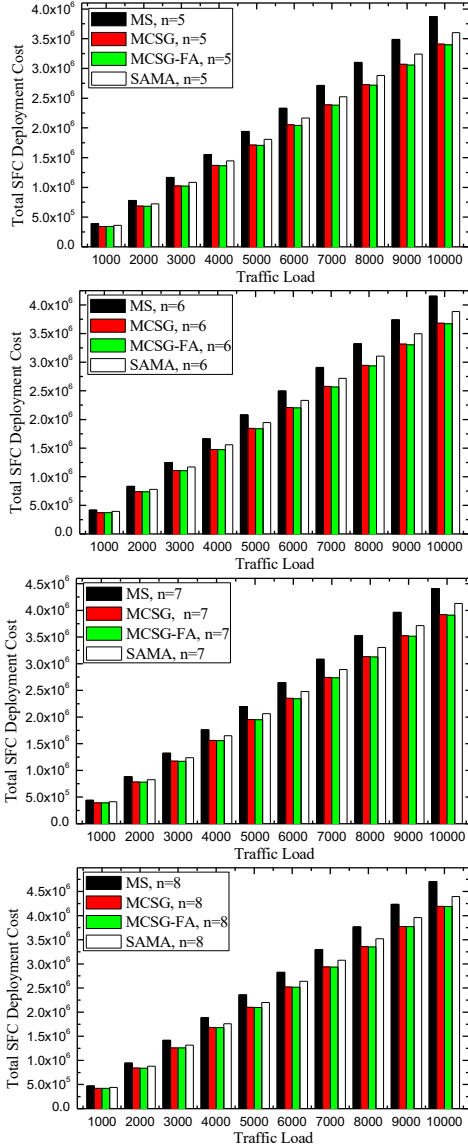


Fig.8 Simulation results of total SFC deployment cost

We compare the total link deployment costs of MS, MCSG, MCSG-FA and SAMA algorithms in Fig.6, compare the total VNF deployment costs of SAMA algorithm and our three algorithms in Fig.7, and compare the total SFC deployment costs of four algorithms in Fig.8. From the results, we can see that the total link deployment cost of MS algorithm is higher than that of SAMA algorithm because MS algorithm pursues the maximal security of the deployment of SFC without considering the deployment cost, whereas SAMA algorithm is designed for minimizing the total placement cost. Hence, the total link deployment cost, the total VNF deployment cost and the total SFC deployment cost of the SAMA algorithm are lower than that of MS algorithm.

Because MCSG algorithm uses the minimal-cost and SSLA-guaranteed strategy for deploying VNFs, it can effectively reduce the link deployment cost, VNF deployment cost and total SFC deployment cost compared to the MS algorithm. The MCSG-FA algorithm first calls MS algorithm to get an initial deployment solution and then tries to find a

deployment solution with the minimal total deployment cost to replace the initial deployment solution. It also can lower the link deployment cost, VNF deployment cost and SFC deployment cost compared to the MS algorithm.

Moreover, in MCSG algorithm and MCSG-FA algorithm, when we deploy VNF  $v_i$  and find the most secure path  $p_{ei}$ , we will find the most secure or minimal-cost  $p^{i+1}(n_j, L_U)$  to reduce the hop of the entire deployment path. Therefore, the MCSG and MCSG-FA algorithms can obtain lower link deployment costs than the SAMA algorithm. We find the path  $p^{i+1}(n_j, L_U)$  and use the minimal-cost strategy, and thus deploy VNF into the entire deployment path with lower cost compared to the SAMA algorithm. Hence, MCSG and MCSG-FA algorithms can get the lower total VNF deployment costs than the SAMA algorithm does. So, the total SFC deployment costs of MCSG and the MCSG-FA algorithms are lower than that of the SAMA algorithm.

## VI. CONCLUSION

In this paper, we investigate the SFC placement problem with SSLA requirement in the federated environment of the cloud-fog networks. To guarantee the security of deployment solution when the security of each physical node and link is given, firstly, we formulate the studied problem as linear programming with SSLA-guaranteed. Then, we propose an algorithm, MS, to maximize the security of the deployment of SFC request. The MS algorithm could maximize the security of SFC deployment, but the total deployment cost is high. To reduce the deployment cost, we design an algorithm, MCSG, to minimize the deployment cost and guarantee the SSLA of deployment. Although the MCSG algorithm can reduce the total deployment cost, it results in a higher blocking ratio. To both improve the blocking ratio and the total deployment cost, we propose another algorithm, MCSG-FA. We validate our proposed algorithms in the cloud-fog networks. The results reveal that our proposed algorithms have better performance than the existing algorithm in the blocking ratio and the deployment cost.

Our future researches will include the integration between the cloud-fog network and AI-based intelligent systems to make our services more robust, secure and efficient.

## ACKNOWLEDGEMENT

This research was partially supported by the National Key Research and Development Program of China (2019YFB1802800), PCL Future Greater-Bay Area Network Facilities for Large-scale Experiments and Applications (PCL2018KP001), VC Research (VCR 0000086) and VC Research (VCR 0000134) of Prof. Victor Chang.

## REFERENCES

- [1] Dan Liao, Yulong Wu, Ziyang Wu, Zeyuan Zhu, Wanting Zhang, Gang Sun, Victor Chang. AI-based Software-defined Virtual Network Function Scheduling with Delay Optimization. Cluster Computing, 22(6), pp.13879-13909, 2019.

- [2] Gang Sun, Dan Liao, Dongcheng Zhao, Zichuan Xu, Hongfang Yu. Live Migration for Multiple Correlated Virtual Machines in Cloud-based Data Centers. *IEEE Transactions on Services Computing*, 11(2), pp.279-291, 2018.
- [3] Sara Khairi, Brahim Raouyane, Mostafa Bellafkih. Novel QoE monitoring and management architecture with eTOM for SDN-based 5G networks. *Cluster Computing*, 23, 1-12, 2020.
- [4] Jian Sun, Yijing Zhang, Dan Liao, Gang Sun, Victor Chang. AI-Based Survivable Design for Hybrid Virtual Networks for Single Regional Failures in Cloud Data Centers. *Cluster Computing*, 22(5), pp.12009-12019, 2019.
- [5] Adel Nadjaran Toosi, Jungmin Son, Qinghua Chi, Rajkumar Buyya. ElasticSFC: Auto-scaling techniques for elastic service function chaining in network functions virtualization-based clouds. *The Journal of Systems and Software*, 152, pp.108-119, 2019.
- [6] Dongcheng Zhao, Dan Liao, Gang Sun, and Shizhong Xu. Towards Resource-Efficient Service Function Chain Deployment in Cloud-Fog Computing. *IEEE Access*, 6(1), pp.66754-66766, 2018.
- [7] Dongcheng Zhao, Dan Liao, Gang Sun, Shizhong Xu and Victor Chang. On Orchestrating Service Function Chains in 5G Mobile Network. *IEEE Access*, 7(1), pp. 39402-39416, 2019.
- [8] Dongcheng Zhao, Dan Liao, Gang Sun, Shizhong Xu, and Victor Chang. Mobile-aware Service Function Chain Migration in Cloud-Fog Computing. *Future Generation Computer Systems*, 96, pp.591-604, 2019.
- [9] Frank Van Lingen, Marcelo Yannuzzi, Anuj Jain, Rik Irons-Mclean, Oriol Lluch, David Carrera, Juan Luis Pérez, Alberto Gutierrez, Diego Montero, Josep Martí, Ricard Masó, Juan Pedro Rodríguez. The Unavoidable Convergence of NFV, 5G, and Fog: A Model-Driven Approach to Bridge Cloud and Edge. *IEEE Communications Magazine*, 55(8), pp.28-35, 2017.
- [10] Chuan Pham, Nguyen H. Tran, Shaolei Ren, Walid Saad, Choong Seon Hong. Traffic-aware and Energy-efficient vNF Placement for Service Chaining: Joint Sampling and Matching Approach. *IEEE Transactions on Services Computing*, 13(1), pp.172-185, 2020.
- [11] Mung Chiang, Sangtae Ha, Chih-Lin I, Fulvio Rizzo, Tao Zhang. Clarifying Fog Computing and Networking: 10 Questions and Answers. *IEEE Communications Magazine*, 55(4), pp.18-20, 2017.
- [12] Fatemeh Jalali, Kerry Hinton, Robert Ayre, Tansu Alpcan, Rodney S. Tucker. Fog Computing May Help to Save Energy in Cloud Computing. *IEEE Journal on Selected Areas in Communications*, 34(5), pp.1728-1739, 2016.
- [13] Ricard Vilalta, Arturo Mayoral, Ramon Casellas, Ricardo Martínez, Raul Muñoz. Experimental Demonstration of Distributed Multi-tenant Cloud/Fog and Heterogeneous SDN/NFV Orchestration for 5G Services. *European Conference on Networks & Communications*, pp.52-56, 2016.
- [14] Mahdi Daghmehchi Firoozjaei, Jaehoon (Paul) Jeong, Hoon Ko, Hyungshick Kim. Security challenges with network functions virtualization. *Future Generation Computer Systems*, 67, pp.315-324, 2017.
- [15] H.J. Jerry Schumacher, Tony Lee, Sumit Ghosh. A Novel, User-Level, Security-On-Demand Paradigm for ATM Networks: Modeling, Simulation, and Performance Analysis. *Journal of Interconnection Networks*, 4(4), pp.429-461, 2003.
- [16] Siyakha N. Mthunz, Elhadj Benkhelifa, Tomasz Bosakowski, Chirine Ghedira Guegan, Mahmoud Barhamgi. Cloud computing security taxonomy: From an atomistic to a holistic view. *Future Generation Computer Systems*, 107, pp. 620-644, 2020.
- [17] Ruben Trapero, Jolanda Modic, Miha Stopar, Ahmed Taha, Neeraj Suri. A novel approach to manage cloud security SLA incidents. *Future Generation Computer Systems*, 72, 193-205, 2017.
- [18] Ori Rottenstreich, Isaac Keslassy, Yoram Revah, Aviran Kadosh. Minimizing Delay in Network Function Virtualization with Shared Pipelines. *IEEE Transactions on Parallel and Distributed Systems*, 28(1), pp.156-169, 2017.
- [19] Long Qu, Chadi Assi, Khaled Shaban. Delay-Aware Scheduling and Resource Optimization with Network Function Virtualization. *IEEE Transactions on Communications*, 64(9), pp.3746-3758, 2016.
- [20] Chen Sun, Jun Bi, Zhilong Zheng, Hongxin Hu. HYPER: A Hybrid High-Performance Framework for Network Function Virtualization. *IEEE Journal on Selected Areas in Communications*, 35(11), pp.2490-2500, 2017.
- [21] Vincenzo Eramo, Emanuele Miucci, Mostafa Ammar, Francesco Giacinto Lavacca. An Approach for Service Function Chain Routing and Virtual Function Network Instance Migration in Network Function Virtualization Architectures. *IEEE/ACM Transactions on Networking*, 25(4), pp.2008-2025, 2017.
- [22] Marcelo Caggiani Luizelli, Weverton Luis da Costa Cordeiro, Luciana S. Buriol, Luciano Paschoal Gaspar. A fix-and-optimize approach for efficient and large scale virtual network function placement and chaining. *Computer Communications*, 102, pp.67-77, 2017.
- [23] Selma Khebbache, Makhlof Hadji, Djamel Zeglache. Virtualized network functions chaining and routing algorithms. *Computer Networks*, 114, pp.95-110, 2017.
- [24] Yong Xiao, Marwan Krunz. QoE and Power Efficiency Tradeoff for Fog Computing Networks with Fog Node Cooperation. *IEEE INFOCOM*, pp.1-9, 2017.
- [25] Gang Sun, Liangjun Song, Hongfang Yu, Xiaojiang Du, Mohsen Guizani. A Two-tier Collection and Processing Scheme for Fog-based Mobile Crowd Sensing in the Internet of Vehicles. *IEEE Internet of Things Journal*, 8(3), pp.1971-1984, 2021.
- [26] Liangjun Song, Gang Sun, Hongfang Yu, Xiaojiang Du, Mohsen Guizani. FBIA: A Fog-based Identity Authentication Scheme for Privacy Preservation in Internet of Vehicles. *IEEE Transactions on Vehicular Technology*, 69(5), pp.5403-5415, 2020.
- [27] Gang Sun, Yijing Zhang, Hongfang Yu, Xiaojiang Du, Mohsen Guizani. Intersection Fog-Based Distributed Routing for V2V Communication in Urban Vehicular Ad Hoc Networks. *IEEE Transactions on Intelligent Transportation Systems*, 21(6), pp.2409-2426, 2020.
- [28] Pengfei Hu, Huansheng Ning, Tie Qiu, Yanfei Zhang, Xiong Luo. Fog Computing Based Face Identification and Resolution Scheme in Internet of Things. *IEEE Transactions on Industrial Informatics*, 13(4), pp.1910-1920, 2017.
- [29] Kai Liang, Liqiang Zhao, Xiaoli Chu, Hsiao-Hwa Chen. An Integrated Architecture for Software Defined and Virtualized Radio Access Networks with Fog Computing. *IEEE Network*, 31(1), pp.80-87, 2017.
- [30] Nicola Iotti, Marco Picone, Simone Cirani, Gianluigi Ferrari. Improving Quality of Experience in Future Wireless Access Networks through Fog Computing. *IEEE Internet Computing*, 21(2), pp.26-33, 2017.
- [31] Zuoxia Yu, Man Ho Au, Qiuliang Xu, Rupeng Yang, Jinguang Han. Towards leakage-resilient fine-grained access control in fog computing. *Future Generation Computer Systems*, 78, pp.763-777, 2018.
- [32] Seongjin Park, Younghwan Yoo. Network Intelligence Based on Network State Information for Connected Vehicles Utilizing Fog Computing. *Mobile Information Systems*, 43(12), pp.1420-1427, 2017.
- [33] Mehdi Sookhak, F. Richard Yu, Ying He, Hamid Talebian, Nader Sohrabi Safa, Nan Zhao, Muhammad Khurram Khan, Neeraj Kumar. Fog Vehicular Computing: Augmentation of Fog Computing Using Vehicular Cloud Computing. *IEEE Vehicular Technology Magazine*, 12(3), pp.55-64, 2017.
- [34] Ricard Vilalta, Arturo Mayoral, Ramon Casellas, Ricardo Martínez, Raul Muñoz. SDN/NFV Orchestration of Multi-technology and Multi-domain Networks in Cloud/Fog Architectures for 5G Services. *Optoelectronics & Communications Conference*, pp.1-3, 2016.
- [35] Ahamed Aljuhani, Talal Alharbi. Virtualized Network Functions Security Attacks and Vulnerabilities. *IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pp.1-4, 2017.
- [36] Konstantinos Fysarakis, Nikolaos E. Petroulakis, Andreas Roos, Khawar Abbasi, Petra Vizarreta, George Petropoulos, Ermin Sakic George Spanoudakis, Ioannis Askoxylakis. A Reactive Security Framework for Operational Wind Parks Using Service Function Chaining. *IEEE Symposium on Computers and Communications (ISCC)*, pp.663-668, 2017.
- [37] Bahman Rashidi, Carol Fung, Elisa Bertino. A Collaborative DDoS Defence Framework Using Network Function Virtualization. *IEEE*

- Transactions on Information Forensics and Security, 12(10), pp.2483-2497, 2017.
- [38] Marco Casazza, Pierre Foulhoux, Mathieu Bouet, Stefano Seci. Securing Virtual Network Function Placement with High Availability Guarantees. arXiv, pp.1-9, 2017.
- [39] Syed Noorulhassan Shirazi, Antonios Gouglidis, Arsham Farshad, David Hutchison. The Extended Cloud: Review and Analysis of Mobile Edge Computing and Fog From a Security and Resilience Perspective. IEEE Journal on Selected Areas in Communications, 35(11), pp. 2586-2595, 2017.
- [40] Rodrigo Roman, Javier Lopez, Masahiro Mambo. Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges. Future Generation Computer Systems, 78, pp.680-698, 2018.
- [41] Mithun Mukherjee, Rakesh Matam, Lei Shu, Leandros Maglaras, Mohamed Amine Ferrag, Nikumani Choudhury, Vikas Kumar. Security and Privacy in Fog Computing: Challenges. IEEE Access, 5, pp.19293-19304, 2017.
- [42] Shuhao Liu, Zhiping Cai, Hong Xu, Ming Xu. Towards security-aware virtual network embedding. Computer Networks, 91, pp.151-163, 2015.